THE NMCSSC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK). ANALYTICAL MANUAL. VOLUME I - DATA BASE PREPARATION. CHANGE 2

Peter K. Cook, et al

National Military Command System Support Center

Prepared for:

Lambda Corporation

21 February 1972

NATIONAL MILITARY COMMAND SYSTEM SUPPORT CENTER

Computer System Manual Number CSM AM 9A-67

21 February 1972

THE NMCSSC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK)

Analytical Manual

Volume I - Data Base Preparation

Submitted by:

Donald F. Webb

DONALD F. WEBB
Major, USAF
Project Officer

REVIEWED BY:

R.E. Harshbarger

R. E. HARSHBARGER
Technical Director
NMCSSC

APPROVED BY:

BRUCE MERRIT
Colonel, USA
Commander, NMCSSC

Copies of this document may be obtained from the Defense Documentation Center, Cameron Station, Alexandria, Virginia 22314.

This document is approved for public release; distribution unlimit. .

CH-2

c2

UNCLASSIFIED
Security Classification

AD/A-006287

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
National Military Command System Support Center (NMCSSC)
Defense Communications Agency (DCA)
The Pentagon
Washington, DC 20301

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
The NMCSSC Quick-Reacting General War Gaming System (QUICK) Analytical Manual, Volume I, Data Base Preparation

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
N/A

5. AUTHOR(S) (First name, middle initial, last name)
NMCSSC: Peter K. Cook
Dianne Hanak
Donald F. Webb
Lambda Corp: Betty J. Ellis
Jack A. Sasseen

6. REPORT DATE
21 February 1972

7a. TOTAL NO. OF PAGES
47

7b. NO. OF REFS
4

8a. CONTRACT OR GRANT NO.
DCA 100-70-C-0065

b. PROJECT NO. NMCSSC Project 631

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
NMCSSC
COMPUTER SYSTEM MANUAL CSM AM 9A-67

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
This document is approved for public release; its distribution is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
National Military Command System Support Center/Defense Communications Agency
The Pentagon, Washington, DC 20301

13. ABSTRACT

This is one of three volumes describing the analytical aspects of the Quick-Reacting General War Gaming System (QUICK). This volume addresses the system data requirements, the organization and structure of the data base, and the concepts and techniques employed to prepare a game data base for subsequent Plan Generation and Simulation. In addition, it identifies the major system limitations and discusses accuracy considerations.

Based upon a suitable data base and user control parameters, QUICK will generate individual bomber and missile plans suitable for war gaming, and simulate the planned events. The generated plans are of a form suitable for independent review and revision. Subsequently, the planned events are simulated; various statistical summaries are produced to reflect the results of the war game. A variety of force postures and strategies can be accommodated.

QUICK is documented extensively in a set of Computer System Manuals (series 9-74) published by the National Military Command System Support Center (NMCSSC), Defense Communications Agency (DCA), The Pentagon, Washington, DC 20301.

PRICES SUBJECT TO CHANGE

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

ia

DEFENSE COMMUNICATIONS AGENCY
WASHINGTON, D.C. 20305

IN REPLY
REFER TO: B225

15 NOV 1974

TO: RECIPIENTS

SUBJECT: Change 2 to Analytical Manual, CSM AM 9A-67, Volume 1, Data Base Preparation

1. This set of change pages documents modifications to the Quick-Reacting General War Gaming System (QUICK) operational at the National Military Command System Support Center. The change set was written under the contract to convert QUICK from the 3800 Control Data Corporation to the 6080 Honeywell. It is designed to eliminate all references to the particular machine on which QUICK is operational, thus making the Analytical Manual machine independent.

2. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.

3. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page. When this change has been posted, make an entry in the Record of Changes.

FOR THE COMMANDER:

J. DOUGLAS POTTER
Chief, Executive Support Office

53 Enclosures
Change 2 pages

EFFECTIVE PAGES - 28 June 1974

This list is used to verify the accuracy of CSM AM 9A-67, Volume I after change 2 pages have been inserted. Original pages are indicated by the letter 0, change 1 by the numeral 1, and change 2 by the numeral 2.

Page No.	Change No.
Title Page	2
ii-vi	2
1	2
1.1	2
2	2
2.1-2.3	2
3-8	2
9-10	0
11	2
11.1	1
12-13	2
14	0
15	1
16-18	2
19	0
20	2
20.1	2
21-23	2
24	0
25	2
25.1	2
26	1
27-30	2
31	0
32-43	2
43.1	2
44-47	2

1c

ACKNOWLEDGMENT

This document was prepared under the direction of the Chief for Development and Analysis, NMCSSC, in response to a requirement of the Studies, Analysis and Gaming Agency (SAGA), Organization of the Joint Chiefs of Staff. Technical support was provided by Lambda Corporation under Contract Number DCA-100-70-C-0065. Technical support for changes 1 and 2 was provided by System Sciences, Inc., assisted by METIS Corporation under Contract Number DCA 100-73-C-0058.

CONTENTS

ILLUSTRATIONS

TABLES

ABSTRACT

This is the first of three Computer System Manuals describing the analytical aspects of the Quick-Reacting General War Gaming System (QUICK). It addresses the system data requirements, the organization and structure of the data base, and the concepts and techniques employed to prepare a game data base to support a specific plan generation/simulation requirement. In addition, applicable constraints and accuracy considerations are described.

QUICK is a two-sided strategic nuclear exchange war gaming system.

It is designed to assist the military planner in examining various facets of strategic nuclear war involving a variety of forces, strategies and starting conditions. Since the volume of data required to support such investigations is substantial, the data base preparation process is designed to provide an efficient means of assembling, maintaining, and organizing an input data base to support user requirements for plan generation and simulation.

The Analytical Manual consists of three volumes. This volume of the Analytical Manual describes data preparation in QUICK. Volume II describes the plan generation process, and Volume III describes the simulation process.

The following is a list of associated documents on the QUICK system.

FUNCTIONAL DESCRIPTION
System Planning Manual SPM FD 90-74
A nontechnical description for senior management personnel

PROGRAM MAINTENANCE MANUAL
Computer System Manual CSM MM 9-74 (five volumes)
Detailed information required for system maintenance and modification

USERS MANUAL
Computer System Manual CSM UM 9-74 (five volumes)

COMPUTER OPERATION MANUAL
Computer System Manual CSM OM 9-74
Instructions and procedures for the computer operators

TECHNICAL MEMORANDUM
Technical Memorandum TM 90-74
Analytical relationships and methodology discussion for users of the system.

CHAPTER 1
INTRODUCTION

GENERAL

The Quick-Reacting General War Gaming System (QUICK) is designed to assist in the study of strategic conflicts involving a large-scale exchange of nuclear weapons. The system is structured into five major subsystems: Data Assembly, Weapon/Target Identification, Weapon Allocation, Sortie Generation, and Simulation. The relationship between these subsystems and the processes of data base preparation, plan generation, and simulation and output processing is shown in figure 1. This first volume of the Analytical Manual describes the process of data base preparation which performs the function of accepting required input data and assembling a game data base which is suitable for input to the plan generation and simulation processes.

Data Base Preparation

The total volume of data required to support a variety of QUICK applications is substantial; however, the majority of these data can be preassembled and maintained in a readily accessible form. This technique is employed by NMCSSC in providing QUICK support. Using the data base preparation process, data are retrieved from various automated data source files maintained by NMCSSC and merged with other manually prepared data to create a source file (called DATADB) for the QUICK system.

This data file may be viewed as a master data base or data library, in that it contains more information than is required to produce a single set of Red and Blue plans. There are no restrictions on the quantity of data that may be maintained in this data file; however, there are constraints (upper limits) on the amount of data the QUICK system can process. While none of these upper limits is considered a significant restriction, they are addressed in greater detail in chapter 2.

Contents of Data Base (DATADB)

To provide a flexible data source, force structures corresponding to established procurement schedules and selected intelligence projections may be maintained for Red and Blue forces. Specifically, the data

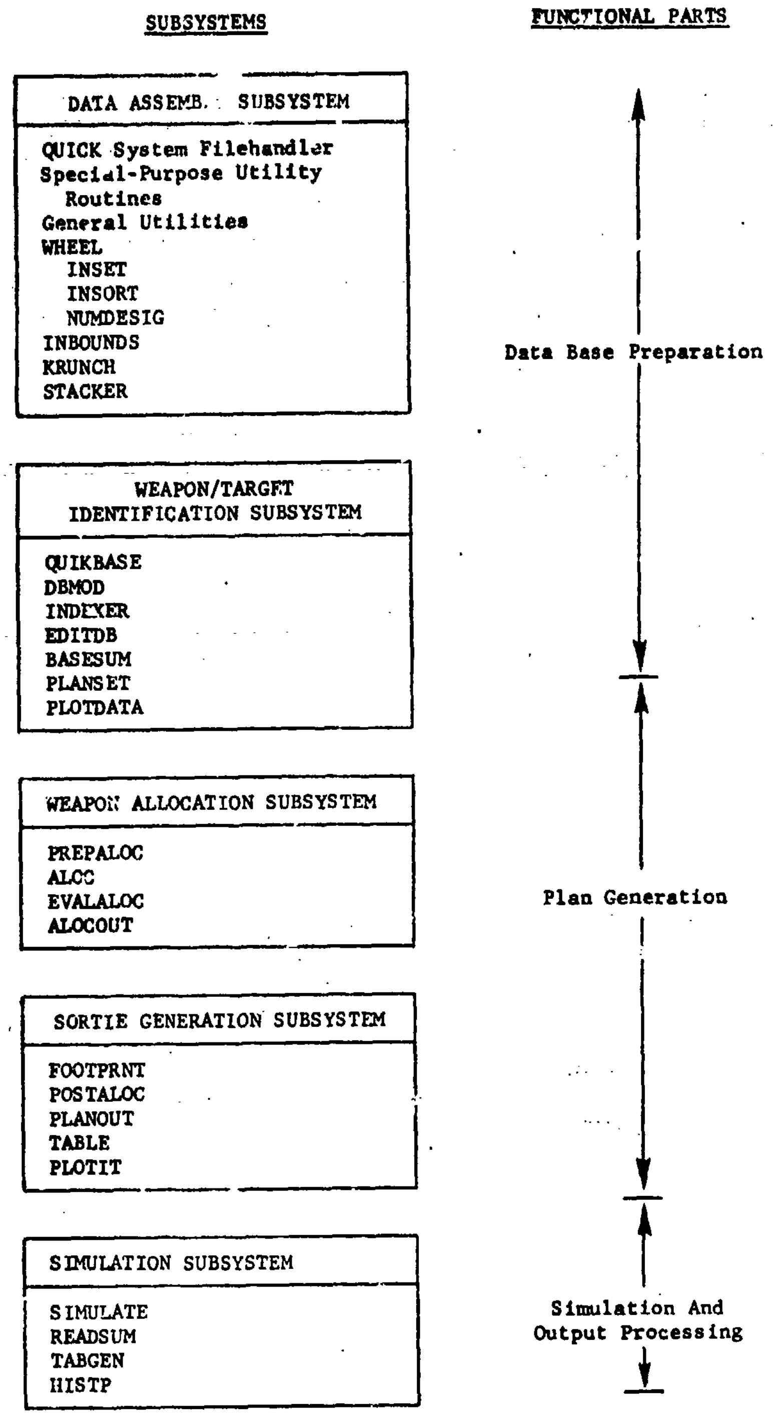

Fig. 1. Major Subsystems of the QUICK System

base contains the information required to define: (1) the capabilities and characteristics of the offensive and defensive weapon systems; (2) the physical characteristics of the installations to be considered as potential targets; (3) related geographic-type data required by the QUICK system for plan generation and simulation, e.g., data describing bomber air defense zones; and, (4) planning parameters such as the estimated probability of destruction before launch (DBL) established for each offensive weapon system (a more specific explanation of the data content and structure of the data base is presented in chapter 2).

Selection of Game Data

The NMCSSC QUICK data base, as described above, contains more data than is required for any single support task. Consequently, the data base preparation process is designed to provide the facilities required to abstract from the data base that information the user desires for a particular simulation or plan development task. In addition, this process provides the capability to add, delete, and/or modify the selected data as required to support the specific game scenario involved. Following selection and modification, a game data base is automatically structured to meet the requirements of subsequent processes and subsystems.

CONCEPT OF OPERATION

QUICK System Overview

The following describes the general concept of operation for the QUICK system and establishes the relationship of the data base preparation process to other major subsystems.

QUICK is structured into five major subsystems as depicted previously in figure 1: Data Assembly, Weapon/Target Identification, Weapon Allocation, Sortie Generation, and Simulation. The principal tasks associated with each of these functional subsystems are summarized below.

- Data Assembly: Assembles and reformats the target data required for a particular plan or simulation.

- Weapon/Target Identification: Selects and processes the Red and/or Blue forces pre-specified for a particular plan or simulation.

- Weapon Allocation: Allocates offensive weapons to selected targets.

- Sortie Generation: Prepares and evaluates missile and bomber attack plans.

- Simulacion: Models and evaluates the significant interactions of opposing war plans developed by QUICK, and prepares summaries and other output data which reflect the results of the simulation.

Figure 2 illustrates the procedure and information flow within the QUICK system. The processing sequence is shown by solid lines, and the information flow by dashed lines. Magnetic computer tape or permanent files may be used to pass information between the various segments of the system.

Although not specifically shown in figure 2, a user/system interactive capability utilizing remote terminals for system operation, is imbedded throughout the system. The resultant terminal interaction permits the user to run jobs at any of several points within any of the subsystems shown.

Processing is initiated by inputting the parameters which identify the potential targets which are to be extracted from the NMCSSC Joint Resource Assessment Data Base (JAD) files. The Data Assembly subsystem reformats this data consistent with QUICK and/or external simulation system specifications. Alternatively, required target data is obtained from an existing, updatable QUICK Data Base stored on a permanent file. Following this, specified Red and/or Blue forces are extracted from the QUICK Data Base and processed by the Weapon/Target Identification subsystem resulting in a Game Data Base which reflects the selected forces and targets.

The next step is to prepare an attack plan for one of the opposing forces. This consists of a force allocation by the Weapon Allocation subsystem, and a detailed set of attack plans prepared by the Sortie Generation subsystem. A single run of each of these subsystems produces a plan for only one side. Consequently, these subsystems must be cycled twice to produce Red and Blue plans.

The major inputs required to initiate this phase of processing are:

- A game data base prepared by the Data Assembly and Weapon/Target Identification subsystems

- A set of parameters which relate to the strategy associated with the plan which is to be developed.

These parameters are supplied by the planner. They reflect his views as to the strategic attack objective, in terms of the relative values of the

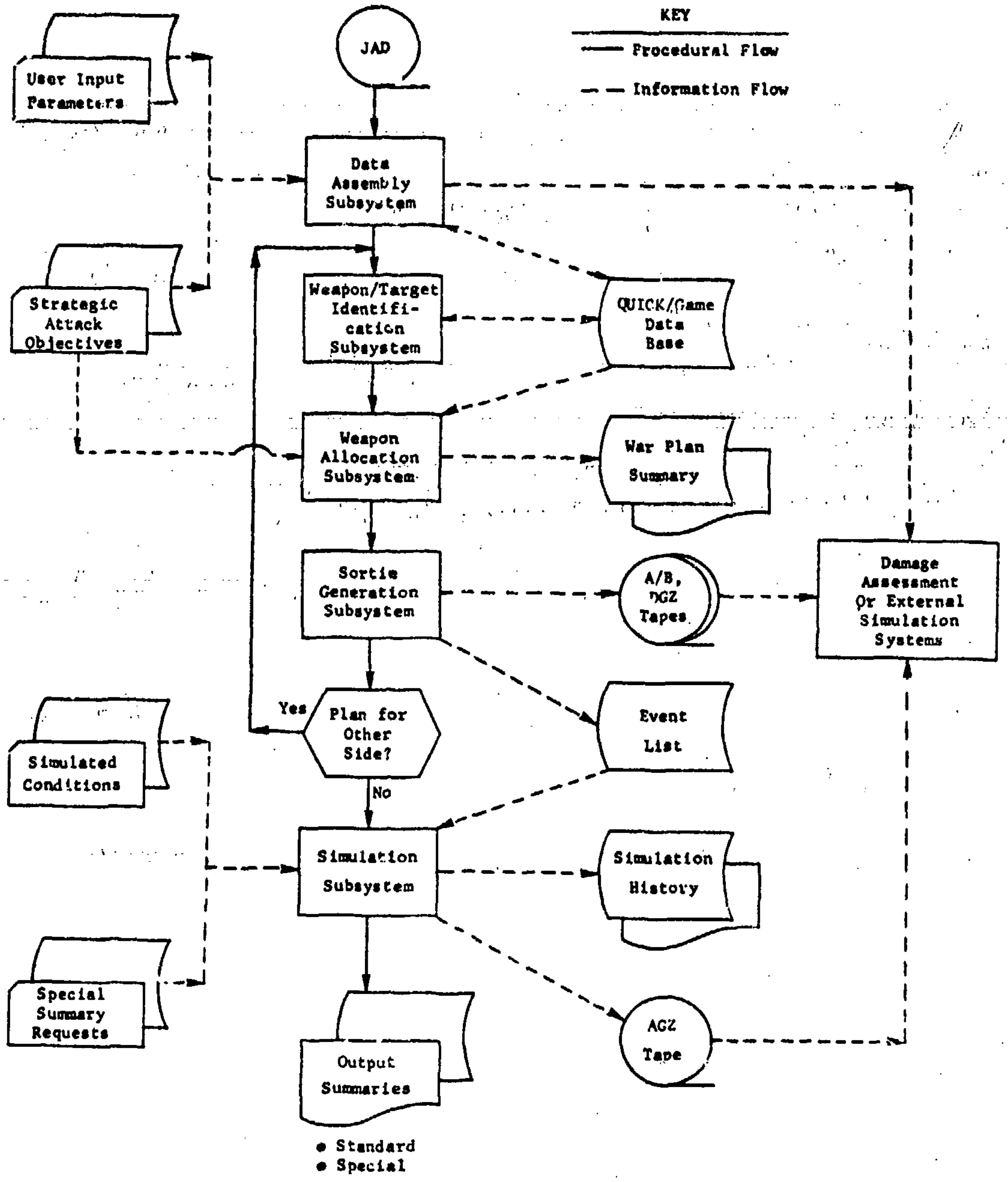

Fig. 2. Procedure and Information Flow in QUICK

various targets being considered, the forces to be withheld, the targeting constraints to be observed, and the initiating force; i.e., which side attacks first.

The target values which are computed on the basis of these parameters reflect in a very significant way the major strategic objectives of the resultant war plan. These values are relative values and are partially contained in the data base itself. QUICK has 15 specific classes of targets. The relative values of the targets contained in any one class are included in the data base; the strategic objectives of the planner who wants to use the plan generation function are expressed in how the value scales of these various classes of targets are related to one another. The user thereby puts more or less relative importance on each of the classes of targets in accomplishing the strategic objectives that he chooses. This, of course, will be related to the kind of strategy he is contemplating for the particular war game, whether a first or second strike, and so forth.

Having established a value for each target, the plan generation phase allocates the weapons (e.g., Red weapons to Blue targets) and prepares the detailed missile and bomber attack plans. If desired, the plans may be printed, inspected, and altered by changing the attack objectives and repeating the process. The series of Red missile and bomber events corresponding to the sortie plan is prepared in a form suitable for input to the Simulator. As a user option, a war plan summary is provided which includes an expected-value estimate of the results of the attack. In addition, the aim point, Desired Ground Zero (DGZ), for each planned weapon can be output for subsequent evaluation utilizing an external damage assessment system. A Blue war plan is then prepared in the same manner as the Red war plan. The system is ready to proceed with the simulation.

The simulation conditions, specifying the starting time for each side and various defense capabilities, are read in from cards or from a remote terminal. The events on the event tapes are then processed in the Simulator along with any new events that are generated. For each event that transpires, a record is made on a history tape or permanent file of all information that might later prove of interest.

When the last event in the game has been simulated, the history tape or file is processed by the Simulation subsystem to prepare the Actual Ground Zero (AGZ) tape which reflects such information as the latitude, longitude, and yield of all successful weapons, and a formatted history which is in a form suitable for game output summarization. The AGZ tape is subsequently processed by a damage assessment system to produce detailed damage assessments. The formatted history is processed by the QUICK Simulation subsystem to provide two outputs: a standard summary of the game, and the results of any special requests for information concerning what transpired during the game.

While the system can proceed automatically through all steps if desired, it may be halted at the end of each subsystem, and the available output inspected for correctness and adequacy. In addition, an interactive capability, utilizing the standard time-sharing subsystems in conjunction with a special-purpose program, permits the user to selectively scan and/or direct the output of individual programs or segments thereof as required.

Data Base Preparation Process (Data Assembly Subsystem)

The Data Assembly subsystem consists of programs which process target data required for plan generation and simulation, and a utility package consisting of programs, subroutines, and functions which perform a variety of support tasks common to several system programs. General descriptions of these functions and their organization in subsequent sections of this volume are provided in the following subsections.

QUICK System Filehandler: QUICK employs a file handling system (subroutine FILEHNR) used by all QUICK programs for tape and disk operations to speed read/write operations.

Special-Purpose Utility Routines: Programs OUTFILE, RELOADF, and DECLARES comprise the special-purpose utility routine package. These have the following purpose:

1. OUTFILE and RELOADF: These programs provide a restart capability used in conjunction with the Weapon Allocation and Sortie Generation subsystems. They enable users to interrupt processing during a subsystem operation and either repeat or resume processing at a later time.

2. DECLARES: This program is used to assist the maintenance programmer or programmer/analyst in writing and maintaining programs which process QUICK data base tapes or files. It places into programs the proper FORTRAN COMMON, EQUIVALENCE and type statements required for the data base being processed.

General Utilities: The programs, subroutines, and functions of the general utilities, which perform a variety of tasks throughout the QUICK system. Among the tasks performed by the general utilities are, for example, error diagnostics, supporting positional and distance calculation functions, tallying of targets, etc.

Data Assembly Programs: The Data Assembly programs consist of programs WHEEL, INBOUNDS, KRUNCH, and STACKER. The data flow for these programs is shown in figure 3.

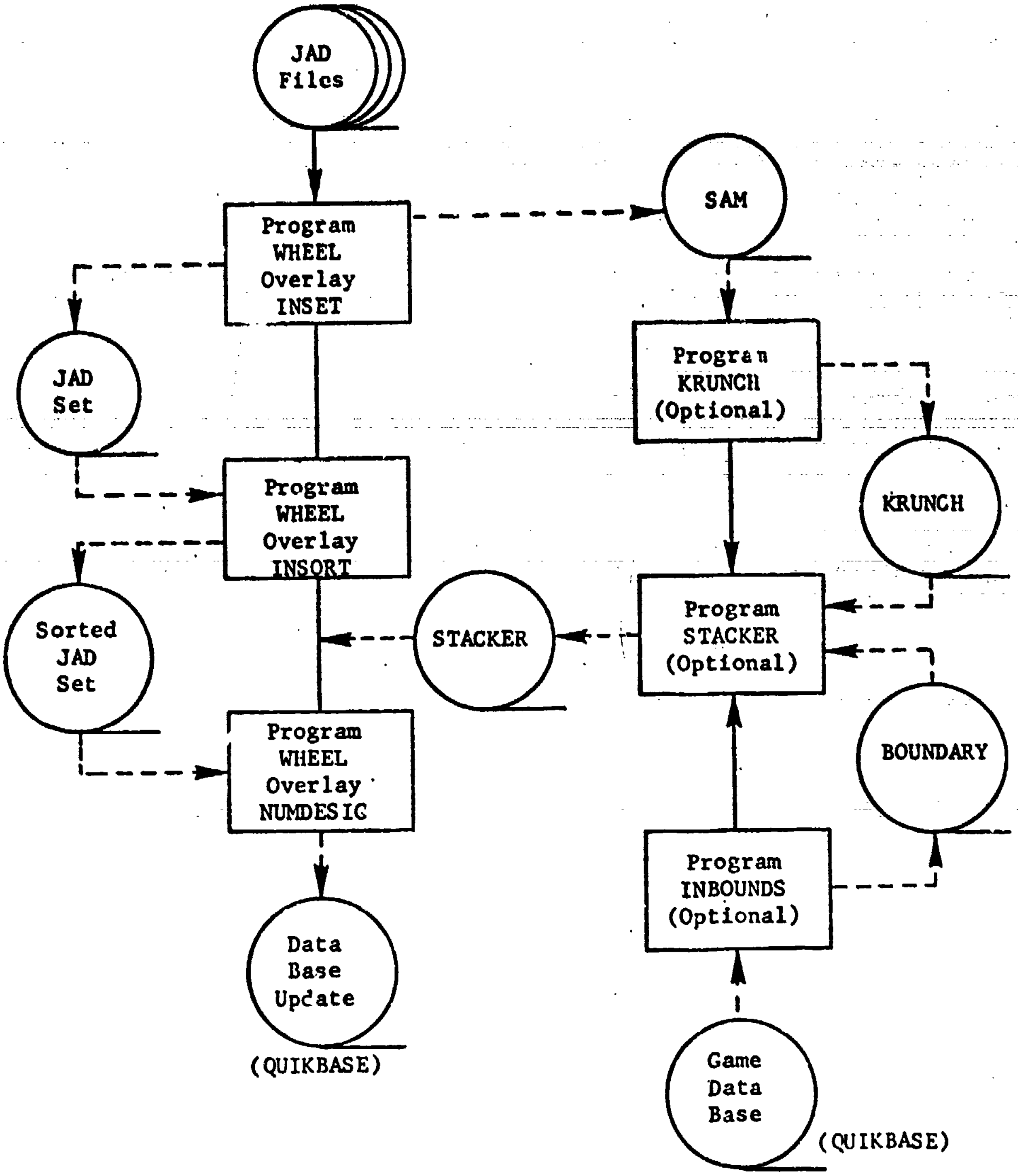
JAD Files
Program WHEEL Overlay INSET
SAM
JAD Set
Progran KRUNCH (Optional)
KRUNCH
Program WHEEL Overlay INSORT
Sorted JAD Set
STACKER
Program STACKER (Optional)
BOUNDARY
Program WHEEL Overlay NUMDESIC
Program INBOUNDS (Optional)
Data Base Update
(QUIKBASE)
Game Data Base
(QUIKBASE)

Fig. 3. Data Assembly Pr Data Flow

The Data Assembly programs extract target items from the JAD files and prepare these items for use in program QUIKBASE of the Weapon/Target Identification subsystem to update an existing data base.

The first program, WHEEL, is a driver program for overlay programs INSET, INSORT, and NUMDESIG. Any one or all of these programs may be selected by executing program WHEEL.

Overlay INSET extracts the desired target item records from the JAD files, and overlay INSORT sorts these item records in a prescribed manner. Overlay NUMDESIG assigns additional information to each item record and prepares the target items for use in program QUIKBASE to update an existing data base.

Three optional programs may be executed to generate data to be used in overlay NUMDESIG, or subsequent programs of the Weapon/Target Identification subsystem for assigning target defense and zone data to item records. These programs are KRUNCH, INBOUNDS, and STACKER. Program KRUNCH prepares the target defense data using the surface-to-air missile (SAM) file produced by overlay INSET. Program INBOUNDS extracts zone data from an existing game data base. Program STACKER combines the outputs from programs KRUNCH and INBOUNDS into a form acceptable to overlay NUMDESIG or subsequent programs of the Weapon/Target Identification subsystem.

Data Base Preparation Process (Weapon/Target Identification Subsystem)

The data base preparation process is a user-oriented data management tool. The process does not involve analytical techniques, per se, but performs data processing functions associated with updating and creating data files used in QUICK.

This process includes the programs, data maintenance routine, and special procedures required to abstract from the QUICK data base (DATADB) the information a user desires for a particular plan or game and assembles it in a form suitable for input to the plan generation, simulation, and output data processes. The process has been designed as a general-purpose, flexible and expandable tool which can be modified to delete and/or add capabilities unique to a specific support task.

The major programs of the data base preparation process are shown in figure 3.1. A brief description of the flow of data and principal functions of the programs involved will provide a basis for understanding the more detailed descriptions which appear later in this volume.

Program QUIKBASE. This program performs the primary function of creating a game data base which defines the general data to be used in the succeeding programs of the subsystem. This program accepts an input data base and, based on user-input parameters, modifies the file to create a

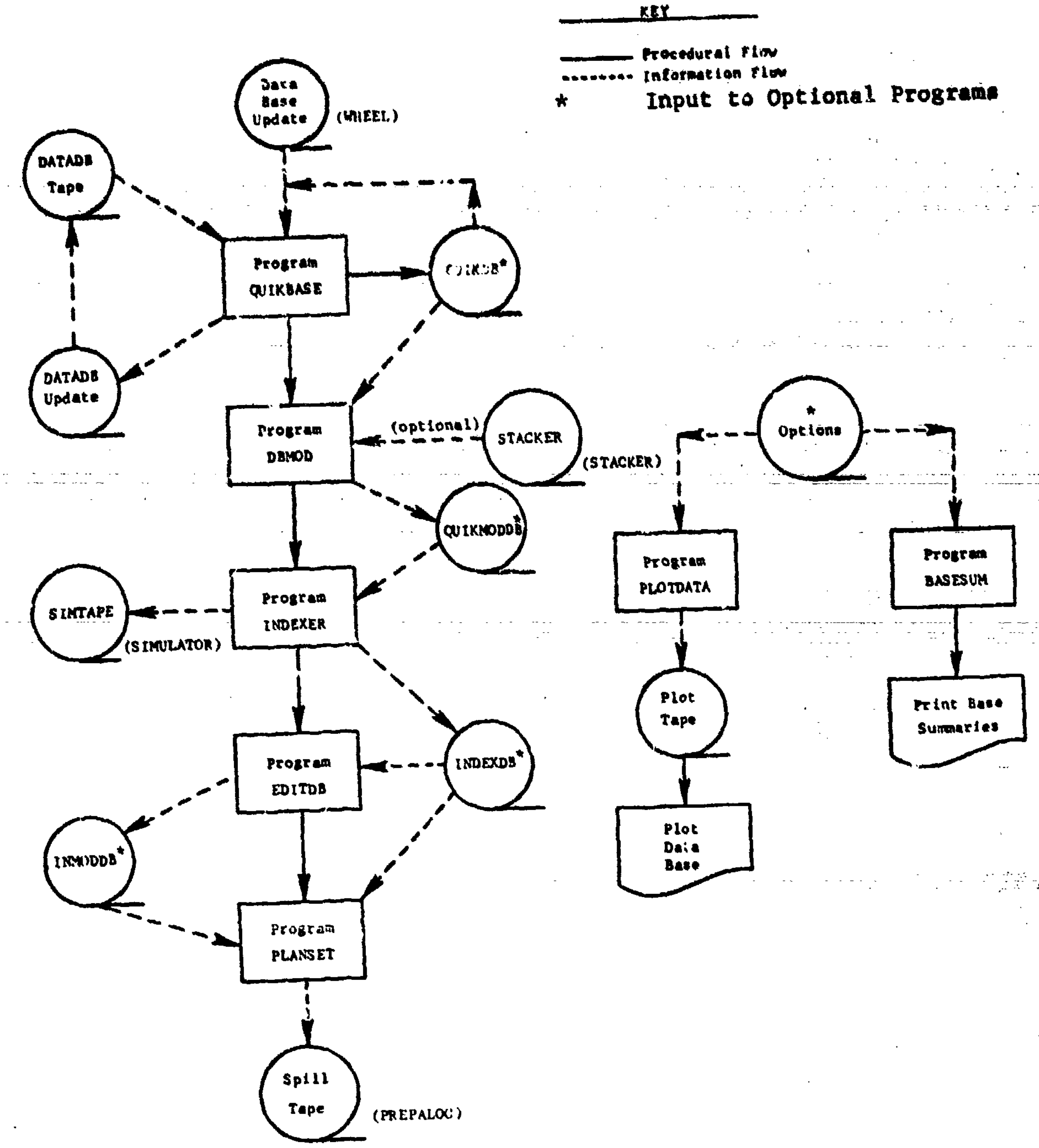

Fig. 3.1. Weapon/Target Identification Subsystem - Data Flow

CH-2

game base file (QUIKDB). The input data base (also called a data library) may be data file DATADB or, as an alternate capability, the data base may be input in card form. On option, the game base file (QUIKDB) can be printed. The program has the additional capabilities of updating an existing data base and creating a damage assessment data base from a game base.

Program DBMOD: This program is designed to perform a specific NMCSSC support task. Its main function is to alter the content or characteristics of a data base in order to adapt it to the specific scenario for which a plan is being developed. As indicated in figure 3.1, this program is utilized after program QUIKBASE to introduce user-desired modifications in the game file. Examples of the types of modifications involved are presented in chapter 3.

Program BASESUM: The purpose of program BASESUM is to summarize a game base file and to print these summaries in tabular form. While figure 3.1 reflects the program operating after program DBMOD, the program may be used to summarize the data base contained on the output tapes produced by programs QUIKBASE, DBMOD, INDEXER, or EDITDB. This program, while not required in the processing cycle, provides a means for checking the input data and is a record of the information contained on any game base tape.

Program INDEXER: To provide for efficient data handling and communications between the programs of the QUICK system, index numbers are assigned to various kinds of data. For example, each item assigned to a target class in the data base (15 target classes are used in QUICK) is assigned an index number, called INDEXN, which may range from 1 to 12,000. For internal coding purposes, such items are referenced by their assigned INDEXN. Program INDEXER is designed to perform the required indexing operations. In addition, target elements which are either exactly collocated or are within close proximity of each other are grouped together to facilitate calculations of target kill probabilities during simulation and to simplify the weapon allocation procedures used in plan generation. These program functions are discussed in greater detail in chapter 3.

The primary NMCSSC input to program INDEXER is the modified data base prepared by program DBMOD (QKMODDB). Program INDEXER prepares two output tapes. The simulation data tape SIMTAPE contains selected weapon and target data and is subsequently input to the Simulation subsystem. In addition, an indexed data base INDEXDB is prepared. The INDEXDB tape serves as input to: (1) the plan generation process and (2) the data output process. If additional modifications are required or data checking is desired, INDEXDB is processed by program EDITDB and a modified indexed game data base INMODDB is output for use in place of INDEXDB.

Program EDITDB: This program is designed to perform editing of the data base tape INDEXDB. It will select or delete items on the basis of country location, and perform consistency checks on certain attributes (including those involved with zone definition). It is (optionally) run following INDEXER.

definitions which apply to consecutive groups of items. This ordering and compression of the file is only an efficiency device which in no way compromises the advantages in flexibility and simplicity of an unstructured file.

This form of data file is processed by a single pass of the entire file, taking appropriate action for each item as it passes through. Most operations, such as counting, extracting, or modifying, are accomplished in this manner.

DATA BASE ORGANIZATION

The QUICK data base consists of two parts, the directory modifications and the Item File. These are described in the following paragraphs.

The Directory

The heart of the Data Base Preparation and Simulation subsystem is the directory, which consists of a list of all the attributes which can be used to describe the items defined in the data base (see Appendix B, QUICK Attribute Names and Descriptions).

It is to be noted that there are three kinds of attributes in the directory:

1. User essential attributes which are necessary if QUICK is to function accurately, e.g., LAT (latitude) item data which must be input by the user

2. Program essential attributes which are automatically generated by QUICK programs, either for internal indexing (e.g., ITYPE) or a result of simulation, e.g., DELAY

3. Nonessential attributes included for the convenience of the user, usually for identification or classification purposes; e.g., WACNO and FLTNO.

The directory of essential attributes is "built into" the program QUIKBASE. The first file of the QUICK data base may consist of modifications to the "built-in" directory. The modifications may consist of changes in the default values of essential attributes, data lists for list checking attributes and additions of nonessential attributes.

This directory facilitates the input of items to the data base, directs various kinds of checking of the attribute values included for each item, and simplifies retrieval and output programs. All items which are input to the data base are checked for consistency. That is, the attributes which are specified to describe an item are checked to determine that they are, in fact, defined in the data base directory and that the attribute value satisfies the range or checklist specifications for those attributes. In addition, the position of attributes in the directory is used as the appropriate index number assigned to the attribute. The index number is stored on the data base file instead of the attribute name to simplify processing.

Directory Conventions

In preparing the directory, certain conventions have been adopted for the units in which values of the attributes are expressed. The ones of greatest interest are: distances expressed in nautical miles; time in hours; and speed in knots. Latitude and longitude are carried within the QUICK system in the following format:

North latitude	0. (Equator) to +90. (North Pole)
South latitude	0. (Equator) to -90. (South Pole)
East longitude	180. to 360. (Greenwich Meridian)
West longitude	0. (Greenwich Meridian) to 180.

Latitude and longitude may be input either using the above format or using the standard degree, minute, second, and direction format. If the latter mode of input is used, the coordinates are converted to the format shown above.

The Item File

After the directory has been formed, any set of attributes can be grouped together and assigned values to define items. The attribute-value pairs are collected to describe such items as targets, weapons, defensive systems, and payloads. The value defined for each attribute of an item is validated by checking the directory.

As previously indicated, a convention which allows "global" attribute definitions may be used in preparing the data base. Within the data base, items are grouped by class and by type within class. Arranged in this manner, groups of items will tend to share common attributes; i.e., a series of items representing B-52 bomber bases may all be assigned the attribute-value pair TYPE=B-52. In this case, the user may define the attribute using a global definition instead of explicitly listing the attribute as an item entry for each bomber base. Once defined, the global attribute remains in effect until: (1) it is removed by an UNDEFINE command inserted in the data base by the user; (2) the target class changes; or (3) the global definition is overridden by a definition contained in the incoming item. In the latter case, the global value is not applied to the item being processed, but it remains in effect for subsequent items in the class.

QUICK Data Classes

The information included in the data base is categorized by CLASS, e.g., bombers, and by TYPE within class, e.g., B-52. Fifteen classes may be used to describe the targetable-type installations included in the data base. The data categories currently associated with each target class are shown in table 3. These classes are identified within the system by referencing the value of the attribute ICLASS, the class number. The class name, attribute CLASS, is not used for internal identification but is included in many of the printouts output by the system. The class names may be changed by the user; however, the program functions unique to a particular ICLASS must be considered in determining the class assignment for targetable items. The attributes assigned to the items of ICLASS 1 through 5 and ICLASS 14 define the item not only as a target but include the attributes which establish its offensive/defensive capabilities and characteristics.

In addition to these target classes, nine auxiliary data classes are used to enter weapon-type data such as the specific composition of a bomber payload (bombs, air-to-surface missiles (ASMs), electronic countermeasures (ECM), and decoys) and geographic-type data required by the system. Geographic-type data are required to define air defense zones and to provide for bomber routing. For example, in plan generation, penetration and depenetration routing of aircraft is controlled through the use of corridors established by the planner. The data defining each of the route points associated with these corridors are included in the auxiliary class segment of the data base. Internally, these classes are identified by the class name shown in table 2 (ICLASS is not assigned to these classes). Hence, these class names may not be changed by the user.

OPERATIONAL RESTRICTIONS

Data Constraints (Upper Limits)

There are no restrictions on the quantity of data that may be maintained in the data library (data base DATADB). There are, however, constraints (upper limits) on the amount of data the QUICK system can process in a single run. These constraints are directly related to the storage requirements outlined above and similar requirements associated with the programs of the other QUICK subsystems. Table 3 provides a list of major constraints associated with the data preparation process.

As shown in table 3, the game data base may contain up to 12,000 items. This includes both Red and Blue items. However, only 5,000 targets per plan can be accommodated. The latter restriction is imposed by the plan generation process and must be considered in forming the data base. For plan generation, target complexes consisting of multiple target class items are treated as a single target; experience with the NMCSSC QUICK data base indicates that at least 20% of the data base items will be assigned to target complexes (criteria explained in chapter 3). Hence, 6,000 target class items can usually be accommodated within the 5,000 target per plan limitation. Considering current QUICK support requirements, none of the existing constraints is considered a significant restriction.

Table 3. Major Input Data Limits

TARGET CLASS DATA	MAXIMUM	CODE*
Target Classes	15	1
Target Types	250	1
Target Types per Class	40/20**	2
Targets (Target Class Items)	12000	1
Targets per Plan (Red or Blue)	5000	2
Targets per Earth Sector	4000	1
Targets - Collocated	4000	1
Targets per Collocation Island	100	-
Target Complexes	4000	1
Target Elements per Complex	40	-
Targets Defended By Terminal Antiballistic Missile Interceptors	500	1
AUXILIARY CLASS DATA		
Warhead Types	50	1
Payload Types	40	2
Air-to-Surface Missile (ASM) Types	20	1
Air Defense Zones (Bomber)	63	2
Command/Control Regions	20	2
Corridors (Penetration)	30	2
Depenetration Points	50	2
Recovery Bases (Bomber) per Depenetration Point	4	-
Refuel Points (User Directed)	20	2
Route Legs	200	2
Route Points	200	2
Weapon Type Characteristics List	40	2
Zone Boundary Legs	200	2

*1 = Total in game base; 2 = Total per side; - = As stated

**Missile and Bomber, 40 each; all others, 20 each

CHAPTER 3
METHODS USED

The data base preparation process is designed to minimize the manual effort required to prepare a QUICK game data base and the associated data files which provide data base information to the subsequent plan generation and simulation processes.

As previously indicated, the process does not involve the use of analytical techniques. Computations performed within the subsystem are not complex. Mathematical operations performed for the purpose of establishing or changing the value of selected attributes involve simple multiplication and division accomplished on the basis of a specific user-input parameter.

The following sections of this chapter provide an explanation of the major steps and functional requirements associated with creating, updating, modifying, and indexing a QUICK game data base.

CREATION OF GAME DATA BASE (QUIKBASE)

The creation of a game base file is performed by program QUIKBASE. It defines the data base to be used by succeeding programs of the QUICK system. This program accepts an input data base (also referred to as a data library) that specifies the attribute-value pairs for each item in the data base. This library (called DATADB) is processed to produce a game base file (called QUIKDB) which contains these attribute-value pairs in a compact format that can be useo easily by the later programs.

Program QUIKBASE serves three functions. First, it provides a capability to create or update a data library tape. This creation or update process may be independent or may include the creation of a game base file (QUIKDB). Second, the program provides facilities to print the game base file (QUIKDB) in a format meaningful to the user. Third, the program can produce a Red and Blue damage assessment data base for use in programs external to the QUICK system.

Six options are available for use in accomplishing the generation of data bases. These options are requested via option cards. Each option is briefly summarized below.

USER OPTION	DESCRIPTION
SETID	Program will read the input file (either cards or BCD card images on tape) and create a data library tape (DATADB) with set and line numbers for each data card image.
SELSTRIP	Program will strip off desired classes by side, or desired sets by line number, and desired attributes from existing data library tapes, perform specified modifications, and place the stripped data on a tape (STRIPD).
UPDATE	Program will update the data library tape (DATADB), create one or more copies of that updated library, and, optionally, create a game base tape (QUIKDB). Input is from cards, a previous DATADB tape, or a STRIPD tape.
QUIKDBG	Program will generate a game base tape (QUIKDB).
PRINTDB	Program will print a QUIKDB tape.
DBASSESS	Program will create both Red and Blue Damage Assessment tapes from the game base tape. These tapes are suitable for input to the SIDAC model.

DATA BASE MODIFICATION AND CORRECTION (DBMOD and EDITDB)

When the QUICK data base is created by merging data retrieved from automated data management systems, some of the data required only for the QUICK system are not present in the data base. In addition, the data base may need to be adapted to the specific scenario for which the plan is being developed. Programs DBMOD and EDITDB alter the content or characteristics of a data base in order to perform this adaptation.

Major modifications and/or additions to the basic game data as required by the QUICK system and the desired plan scenario are performed by DBMOD. EDITDB is used for minor plan variations when the planner does not wish to disturb the basic plan indexing scheme, and/or wishes the data base to be checked to see whether the values of attributes are consistent.

The major data augmentation tasks performed by program DBMOD are:

1. Targets which are inappropriate for the plan under consideration; i.e., those targets assigned the attribute RESERV=0, are excluded from further consideration

2. The appropriate number of weapon vehicles per bomber squadron (NOPERS) is chosen, depending upon the particular plan being developed (retaliatory, initiative, or surprise)

3. The number of bombers in commission (NOINCO) for each squadron is calculated by specifying that NOINCO is equal to a user-specified fraction of NOPERS

4. The number of bombers which are on alert (NOALER) for each squadron is calculated by specifying that NOALER is equal to a user-specified fraction of NOINCO

5. The appropriate relative effectiveness (EFECTN) of each interceptor squadron and each defensive command/control installation is selected based on the type plan being developed

6. The relative value (VAL) of urban/industrial targets is calculated as a function of general industrial worth (IGIW) and population (POP) according to the formula VAL = A·IGIW + B·POP, where A and B are player inputs

7. If required, target defenses (TARDEFs) are processed

8. If required, air defense zones are calculated for classes 4 and 5. These zones are usually already part of the data base prepared for QUICK.

For input, the plan generation process uses subsets of the data that are contained in the general data base. The exact composition of these subsets depends upon which type of plan is being developed. Thus, there is a need to examine in detail all of the attributes contained in the basic game data base and, depending upon which of the three possible plans is to be constructed (retaliatory, initiative, or surprise), to omit from further consideration those data which are not relevant.

This task is accomplished by running program DBMOD after program QUIKBASE. The program sequentially examines each item in the data base file. According to user-specified input parameters, items with the attribute RESERV equal to zero are omitted; selection is made of the correct number per squadron, number in commission, number on alert, and effectiveness; and adjustment is made of the attribute VAL as a linear combination of IGIW and POP.

Program EDITDB examines items in the indexed data base and deletes items on the basis of their country location code, CNTRYL. The user can thus provide for variations in the target system by country without disturbing the basic indexing scheme. It also checks attributes that should hold certain relationships with others.

SUMMARIZING THE GAME DATA BASE (BASESUM)

Program BASESUM summarizes game data bases and prints these summaries in tabular form. Program BASESUM may be used to summarize the data base contained on the output tapes produced by programs QUIKBASE, DBMOD, INDEXER, or EDITDB. The only input required by the program is a data base tape. The output consists of printed summary information. There are no user-input parameters required for this program.

Program BASESUM makes two passes through the data base and summarizes the contents by side and class. For each side-class combination, one table is generated in which the colums are the types found and the rows are all the attributes defined for this class. One line called ITEMS is added to the row heading, although it is not a true data base attribute. It contains the count of the number of items of that side, class, and type.

SUBSYSTEM INTERFACES WITH WEAPON/TARGET IDENTIFICATION SUBSYSTEM (INDEXER)

In order to provide for efficient data handling and communication between the programs of the QUICK system, it is necessary to assign indices to the data in the data base. Program INDEXER of the Weapon/Target Identification subsystem is designed to perform this important function.

The input to program INDEXER consists of either the game data base as created in program QUIKBASE (QUIKDB) or the modified data base (QKMODDB) as constructed in program DBMOD. The output from the program consists of an indexed data base tape, INDEXDB, which provides data used in plan generation, and a simulation data tape, SIMTAPE, which is used in program SIMULATE of the Simulation subsystem.

The indexed data base contained on INDEXDB is normally input directly for plan generation. However, if required, additional modifications to this file may be implemented using program EDITDB. If this option is exercised, INDEXDB becomes an input to program EDITDB which prepares a modified indexed data base INMODDB for input to plan generation.

The data base on the INDEXDB tape differs from the data base which is input to INDEXER (QUIKDB or QKMODDB tape) in the following ways: the indices INDEXNO, ITYPE, and JTYPE are defined for all appropriate items, and a positive value of the attribute ICOMPLEX is assigned to all elements of the target complexes formed by INDEXER.

The SIMTAPE is a library type file. It provides information which supplements the war plan data provided by plan generation and is required by the Simulator to model and evaluate the results and interactions of the planned events. The SIMTAPE information includes:

1. The array COLAR which provides information on collocated targets, e.g., the index number INDEXN and the relative coordinates (offset distances in units of .02 nauticel miles) between collocated targets.

2. The array STATUS which provides information on every target class item processed by INDEXER. This array consists of one word for each value of INDEXN. A series of codes and indices packed within each word provides: (1) information on the target (INDEXN item), e.g., its collocation status (one if collocated, otherwise zero); and (2) indices to other arrays containing target information, e.g., the index to the array VULN which contains the target's vulnerability number.

3. Several tables which provide (1) type characteristics, i.e., data such as probability of a launch abort, which are the same for every vehicle of a given type and (2) data establishing the defensive capabilities of each side, e.g., the defensive potential of each bomber air defense zone.

Indexing Operations

To facilitate cross-referencing between data base items, all items in the target classes are assigned indices. Each item is assigned a unique index number (attribute INDEXN) and two indices, ITYPE and JTYPE, which identify the item as a member of a specific "type set."

Program INDEXER assigns indices to the various kinds of data as follows. For all classes that contain items which could be considered to be targets, the user assigns a value of the attribute ICLASS from 1 to 15 at the time the data base is prepared. All target types (attribute TYPE) which belong to these classes are assigned distinct values of the attribute ITYPE, and all types within each class are assigned distinct values of the attribute JTYPE. Each item in classes 1 through 15 is then assigned a distinct value of the attribute INDEXN; this assignment preserves the partial ordering of the items due to the ITYPE assignment.

The assignment of these indices is performed by program INDEXER which makes several passes through the game data base file. In pass one, the values of ITYPE, JTYPE are assigned, and other indexing operations are completed as follows.

To facilitate cross-referencing in subsequent programs, program INDEXER constructs a set of breakpoint tables which reflect the indexed structure of the data base. With the aid of these breakpoint tables, which give the beginning indices INDEXN of each class and type and the number of RED and BLUE types in each class, it is possible to obtain the class, type, and side of any item from its index number. The breakpoint tables are included on the output tapes prepared by program INDEXER (INDEXDB and SIMTAPE).

Collocation Islands and Complex Targets

In order to allow more efficient operation of the plan generation and simulation processes, the individual target items in the game data base are aggregated into "clusters." These clusters may take two forms, collocation islands or complex targets.

Collocation Islands: To facilitate the calculation of target kill probabilities during simulation, the QUICK design provides for grouping appropriate targets into collocation islands. Collocation islands are defined by the following criterion: if the distance between two targets is less than the sum of the lethal radii of an input weapon yield (based on sides), considering the hardness of each target, the targets belong to the same collocation island. A collocation island consists of all targets (up to 100) which are linked by this distance criterion. In practice, islands are usually rather small clusters. Targets are said to be collocated if they belong to the same collocation island; a collocated target is one which belongs to some collocation island. During simulation, if a weapon is successfully delivered against a collocated target, the effects of the burst are assessed against all targets in the collocation island. Data reflecting the relevant target-to-target spacing for each collocation island are referenced by the Simulator through the use of the index number INDEXN assigned to each target.

Complex Targets: The definition of a complex target is identical to the definition of a collocation island, except that the required distance between targets is one-half the distance defined for collocation. Therefore, every complex target is a subset of some collocation island. Thus, complex targets consist of target elements (up to 40 data base items) which are either exactly collocated or within the defined distance. Under this criterion, the target elements are reasonably close together and should be considered as a unit. As a result, the Plan Generator processes a complex target as a single simple target during the weapon allocation phase (see Analytical Concepts and Techniques, Target List Preparation, in chapter 2, Analytical Manual, Volume II).

Multiple Targets: A multiple target is defined as several independent, identical missile targets (such as separate missile silos in a Minuteman squadron) that are close together relative to the range of the weapon systems, but far enough apart so that each target element is treated as an independent aim point. For such targets, the right targeting for one of them is undoubtedly the right targeting for them all. Thus, plan generation determines the targeting of all elements of a multiple target through a single calculation of targeting for a representative target (of the appropriate multiplicity).

The search for collocation begins by comparing the first element in the list with the following elements if the difference in longitude is sufficiently small, the actual distance is calculated and compared with the sum of the critical distances for the two targets. When two targets are found to be collocated, COL and CL are set to 1 for both, and CLT is set to 1 for the second. If they are sufficiently close to be elements of a complex target (one-half the distance for collocation), CP is set to 1 for both, and both indices are entered in LCOMP. When a difference in longitude is encountered which is too great, processing of the item being investigated is considered finished, and CLT is set to 0 for that item. If the item is a member of a complex target, the next member of that complex in the list LCOMP is compared in the same way with all other items to find additional members of the complex; this process is repeated until all items in LCOMP have been investigated, and the complex is finished. The complex is assigned a value of ICOMPLEX which is packed along with IND(J), i.e., INDEXN, into consecutive words in the array COMPLEX. Next, LCOMP is cleared and investigation of the current collocation island is continued, beginning with the next item for which CLT is 1. When the collocation island is finished, the required data for the items which belong to the island are packed in the array COLAR, CL is reset to 0, and COLAR is written on disk. Then the next item in the list for which COL is 0 is compared with all others to restart the investigation. When the list is exhausted, all collocation islands and complex targets for the current earth sector have been processed. Subsequent earth sectors are processed in the same manner until all data base target items have been checked.

During the last pass through the data base, as each item is processed, those which are elements of complex targets (as indicated by the flag COMP described above) are assigned the value of ICOMPLEX and written on the INDEXDB tape. The complex target data are used in plan generation (see Analytical Concepts and Techniques, Target List Preparation, in chapter 2, Analytical Manual, Volume II). The collocation data contained in the COLAR array are subsequently written on the SIMTAPE for use during the simulation process.

Preparation of Simulation Data

The data base information required by the Simulator is transmitted via the SIMTAPE prepared by program INDEXER. The SIMTAPE contains only that data required for simulation (a significant segment of the data included in the data base is required in plan generation but not used by the Simulator). These data are organized as a series of indexed tables (arrays) to enhance storage and retrieval during the simulation phase. On option, the contents of each SIMTAPE table may be printed.

THE CONTENTS OF THIS PAGE INTENTIONALLY DELETED.

CHAPTER 4
ACCURACY

The programs in the Data Assembly subsystem perform mainly data processing functions wherein computational accuracy is not a likely source of error. None of the calculations which are performed in this subsystem are complex. System computer word structure is large enough to contain all of the significant digits necessary for proper execution of the system. The only problems which may affect the operations of these components with respect to the remaining programs in the system are input and output functions. These are discussed below.

Input

Inputs to program QUIKBASE of the Weapon/Target Identification Subsystem do present an operational problem with respect to accuracy. The program now accepts data in the first eight columns of each 10-column field in a card image. This means that all attributes and their assigned values must be represented in eight characters or less.

Since many of the quantities (values of attributes) are included in data tapes prepared for subsequent processing, the level of significance on the accuracy results from expressions using these attributes will be correspondingly limited. This is not believed to represent a significant limitation on the capabilities of the QUICK system.

Output

The data output from the Data Assembly subsystem and the various programs internal to the subsystem will be as accurate as that data input to program QUIKBASE.

The Data Assembly subsystem provides listings of data which have been processed by the subsystem. These listings reflect the stored values of attributes at the time they are printed. In some cases, when a real number (floating point) exceeds the expected output range (the format indicated in the directory portion of the Data Base File) the program in which the print is requested indicates that condition by placing all asterisks in the output field. Thus, the presence of asterisks in the output indicates incompatibilities between the data being printed and the format used in printing the data. This in no way reflects a deviation of accuracy in the data quantities. If an integer (fixed point) number exceeds the expected output range, the computer operating system software removes the most significant digits (including sign) from the output field with no indication. For example, if the number 12345 were to be printed under FORTRAN format I2, the digits 45 would appear in the output listing. In many cases the format in the directory, for example, can accommodate only eight characters, a limitation on the input procedures of the Data Assembly subsystem. This limits the number of characters which can be printed but does not limit the quantity in memory or on the data files being processed or generated.

FLAG: A code used in imposing restrictions on the allocation of weapons within QUICK.

General War: Armed conflict between major powers in which the total resources of the belligerents are employed, and the national survival of a major belligerent is in jeopardy.

Input: Any factors, data, parameters, values, or instructions required for proper operations of a model or submodel to produce game results.

JAD: Joint Resource Assessment Data Base. The JAD is an automated repository of information acquired from several sources which is stored and maintained by NMCSSC.

Library Tape, QUICK System: A magnetic tape on which the programs and data handling routines of the QUICK system will reside.

Magnetic Tape: A computer storage device in which data are stored in the form of magnetic spots on metal or coated plastic tape.

NEMO Model: The Nuclear Exchange MOdel maintained by the Navy for simulation of a two-sided global nuclear war.

Nuclear Vulnerability Assessment: The estimation of probable or expected effects of hypothetical nuclear attacks on population, forces, and resources.

Posture: Relative place or position; state or condition at a given time, especially in relation to other persons or things.

Probability of Damage (PD): The probability that damage will occur to a target expressed as a percentage or as a decimal.

RISOP: A hypothetical Red Integrated Strategic Offensive Plan.

SIDAC: Single Integrated Damage Analysis Capability System.

SIOP: The Single Integrated Operational Plan.

TASK: A two-character descriptive code assigned to all targets.

APPENLIX B
QUICK ATTRIBUTE NAMES AND DESCRIPTIONS

This appendix lists, in alphabetical order, the attributes used in the NMCSSC QUICK data base. Also provided are the definition/description of each attribute as it pertains to the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
ABRATE	Probability of aircraft in-flight abort per hour of flying time
ADBLI	ALRTDB probability for initiative attack
ADBLR	ALRTDB probability for a retaliatory attack
ADEFCM	Area ballistic missile defense (BMD) component index (radar or missile launch site)
ADEFZO	Area ballistic missile defense (BMD) zone number
AGX*	Offset X-coordinate of AGZ (50ths of nautical miles)
AGY*	Offset Y-coordinate of AGZ (50ths of nautical miles)
AHOB*	Actual height of burst of weapon (air or ground)
ALRTDB*	Probability of destruction before launch (DBL) of alert delivery vehicle (missile or bomber)
ALRTDL*	Delay of alert vehicle before commencing launch (hours)
AREA	Area of a bomber defense ZONE (millions of nautical miles^2)
ASMTYP	Air-to-surface missile type
ATTRCO	Attrition parameter for a bomber corridor (probability of attrition per nautical mile)
ATTRLE	Attrition parameter for each route leg in bomber sortie (probability of attrition per nautical mile)
ATTRSU	Amount of original attrition that remains after defense suppression
AZON1	First area defense zone covered by a BMD long-range radar

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
AZON2	Second area defense zone covered by a BMD long-range radar
AZON3	Third area defense zone covered by a BMD long-range radar
BCODE*	Code indicating the outcome of a simulated bomber event
BLEGNO	Index to boundary line segment
CEP	Circular error probable (CEP), delivery error applicable to gravity bombs, air-to-surface missiles, and missile warheads (nautical miles)
CLASS	Class name assigned to identify sets of TYPES in data base
CLASST*	Target CLASS
CNTRYL	Country code for country where item is located. Items can be deleted by CNTRYL in NUMDESIG or DBMOD.
CNTRYO	Country code for country which owns the item
CNTYLO*	Target country code for country where the target is located
CNTYOW*	Target country code for country which owns the target
CODE*	Outcome code for a general event used in simulation
CPACTY	Capacity of a bomber recovery base (number of vehicles)
DATEIN	Earliest date in inventory (year)
DATEOU	Latest date in inventory (year)

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
DEFRAN	Typical range of interceptors at defense bases near a corridor (nautical miles)
DELAY*	Delay time (e.g., launch delay time) (hours)
DESIG	Target designator code, e.g., AB100, which uniquely identifies each target element included in the data base
DGX*	Offset X-coordinate of desired ground zero (DGZ) (50ths of nautical miles)
DGY*	Offset Y-coordinate of DGZ (50ths of nautical miles)
DHOB*	Height of burst of weapon (0-ground, 1-air)
EFCNS1 EFCNS2	Attributes assigned to fighter interceptor units (ICLASS = 5 in the data base): the value EFCNS1 or EFCNS2 is assigned to the attribute EFECTN depending on value of DBMOD input parameter POSTUR (if POSTUR = 1, EFCNS1 is used; otherwise, EFCNS2 value is assigned)
EFECTN*	Air defense capability (arbitrary scale) established by user to indicate relative effectiveness of air defense command and control installations and fighter interceptor bases
EVENT*	Index to event tape
EVENTN*	Index to type of event which did not occur
FFRAC	Fission fraction (fission yield/total yield)
FLAG	Numeric cod (1 through 9 permitted) used to impose restrictions on the allocation of weapons within QUICK
FUNCTI	Operational application code for a weapon system (e.g., ICBM)
FVALH1	Fraction of value of target in first hardness component

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
FVALT1 FVALT2 FVALT3 FVALT4 FVALT5	Fraction of target value remaining at T1, T2, T3, T4, and T5, respectively
H1	First hardness component of a target (VULN)
H2	Second hardness component of a target (VULN)
HILOAT	The ratio of the low-altitude attrition rate to the high-altitude rate (decimal fraction)
IALERT	Alert status; 1 = alert, 2 = nonalert
IALT*	Altitude index (1 = high, 0 = low)
IATTAC	Selection index for preferential area BMD; 1 forces target selection for defense
ICLASS	Class index assigned for game
ICLSST*	Target class index
ICOMPL*	Complex index
IDBL	Index to data tables for time-dependent destruction before launch probability
IDUD*	Dud warhead indicator; assigned to weapons which arrive at the target but fail to detonate; 1 = dud warhead
IGIW	Indices of General Industrial Worth (IGIW) (dollars)
IGROUP*	Group index assigned for weapon grouping during game
IMIRV	Identifying index for system with multiple independently targetable reentry vehicles

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
INDEXN*	Index of a data base item (potential target) used during processing to identify the item
INDV*	Vehicle index within base
INTAR*	Target index (corresponds to INDEXN)
IPENMO*	Penetration mode; 1 = aircraft uses penetration corridor, 0 = penetration corridor not used
IPOINT	Index to a geographic point
IRECMO	Recovery mode; 1 = aircraft should plan recovery, 0 = aircraft recovery not planned
IREFUL	Bomber refueling code
IREG	Index to identify a geographic region
IREP	Reprogramming index (capability of missile squadron)
ISITE	Site number
ITGT*	Target index number
ITIME	Index to time periods in time-dependent DBL data tables
ITYPE*	Type index assigned for game
ITYPET*	Target type index
IVULN*	Index to vulnerability number table
IWTYP2	Second warhead type
JCLASS	Used for items in the WTCL auxiliary class. The value will be the class number (ICLASS) of the weapon type defined in the WTCL item

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
JTYPE**	Type index within class
JTYPET**	Target type index within class
KORSTY	Parameter to adjust mode of corridor penetration
LAT	Latitude (degrees)*
LCHINT	Time (in minutes) between successive vehicle launches from the same base (missile or bomber) subject to the simultaneous launch condition
LEGNO	Index to line segment
LINK	The index of a leg linked to the current point
LONG	Longitude (degrees)*
MAXFRA	Maximum value of weapon resources to be used relative to target value (in processing MAXCOST = MAXFRA)
MAXKIL	Desired maximum damage expected for a target
MCODE**	Code indicating outcome of simulated missile event
MINKIL	The required minimum damage established for a target

*Latitude and longitude are carried internally in the QUICK system in the following format:

North latitude	0.	(Equator) to +90 (North Pole)
South latitude	0.	(Equator) to -90 (South Pole)
East longitude	180.	to 360. (Greenwich Meridian)
West longitude	0.	(Greenwich Meridian) to 180.

These attributes may be input in either the above format or in standard degree, minute, second, direction format.

**Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
MWHDS*	Number of missile warheads penetrating area defenses to terminal defense
NADBLI	NLRTDB for initiative attack
NADBLR	NLRTDB for retaliatory attack
NAINT	Number of area ballistic missile interceptors at an interceptor launch base
NAME	Arbitrary alphameric descriptor for any item included in the data base
NAREAD	Number of decoys per independent reentry vehicle for area BMD
NASMS	Number of ASMs carried by a bomber
NCM	Number of countermeasures carried by vehicle
NDECOY	Number of decoys on a bomber or number of decoys per independent reentry vehicle for terminal BMD
NDET*	Number of warheads detonating in current event
NEXTZO	The adjacent zone to a side of a defense zone
NLRTDB	Delay of nonalert vehicle before commencing launch (hours)
NLRTDL*	Probability of destruction before launch (DBL) of nonalert vehicle
NMPSIT	Number of missiles per site
NOALER	Number of vehicles on alert at a base
NOBMB1	Number of first bomb type carried by vehicle
NOBMB2	Number of second bomb type carried by vehicle
NOINCO	Number of delivery vehicles in commission
NOPERS	Number of weapon vehicles per squadron

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
NPEN*	Number of warheads penetrating in current event
NPRSQ1 NPRSQ2 NPRSQ3 NPRSQ4	Attributes used in program DBMOD to compute the value of the attribute NOPERS for bomber units; numbers 1, 2, 3, and 4 specify surprise initiative retaliatory and other attack plans, respectively
NTARG*	Number of targets in missile launch event
NTINT	Number of terminal BMD interceptors at target
NUMDBL	Number of aircraft destroyed before launch
NWHDS	Number of warheads per independent reentry vehicle (missiles)
NWPNS*	Number of weapons in a group
NWTYPE*	Warhead type
PARRIV	Probability of bomber arrival in current event
PAYALT	Bomber payload release altitude
PAYLOA	Index which identifies entire weapon and penetration aid complement on a vehicle
PDES	Probability that launch failure destroys missile
PDUD	Probability a warhead will fail to detonate
PFPF	Probability of failure during powered flight (missiles)
PINC	Probability that a missile is in commission
PKNAV	Single shot kill probability of a weapon against a naval target (a value greater than zero restricts weapon use to naval targets)
PLABT	Probability of vehicle launch abort
PLACE*	Index to geographic location of an event
PLACEN*	Index to geographic location of an event which did not occur

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
POP	Population (cities) (thousands)
POSTUR	Currently not used as an attribute
PRABT	Probability of refueling abort
PRIMET	Primary target indicator for a weapon
PSASW	Destruction before launch probability assigned a weapon for a specified time period
RADIUS	Size descriptor for area targets (nautical miles)
RANGE	Vehicle range (nautical miles)
RANGED	Range decrement for low-altitude aircraft flight (high range/low range)
RANGER	Range (nautical miles) of bomber with refueling
REL	Reliability - probability that weapon system will arrive at target given successful launch
RESERV	Technique used to remove certain targets from weapon allocation when RESERV = 0
RNGMIN	Minimum range (nautical miles) for the missile type. Used in computing flight times for missiles
SIDE	Item side name, currently either "RED" or "BLUE"
SIMLUN	Maximum number of vehicle launches which can occur simultaneously from one base
SPDLO	Speed at low altitude (knots)
SPEED	Speed (knots)
T1 T2 T3 T4 T5	Times of departure of first through the fiith value components of a target
TAIM*	Number of aim points perceived by terminal defense in current event

*Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
TARDFH	Level of local bomber defense at high altitude[*]
TARDFL	Level of local bomber defense at low altitude[*]
TASK	Target task code indicating targeting priority
TGTSTA	Indicates target status as dynamic or nondynamic; in simulation status (alive/dead) is maintained for dynamic targets
TIME[**]	Game time at which event occurred (hours)
TIMEN[**]	Time planned for event which did not occur (hours)
TMDEL	Mean delay time to relaunch after a nondestructive aircraft abort (hours)
TOFMIN	Minimum flight time (minutes) for missile types. Used in computing flight time for missiles
TPASW	Time at which a time period ends for DBL data tables; there may be up to 10 time periods for each table.
TRETAR	Time required to retarget for known in-flight missile aborts (hours)
TTOS	Total time on station (for a tanker) (hours)
TVUL	Time a missile remains within vulnerable range of launch site (hours)
TYPE	Arbitrary alphameric designator (type name) to identify smallest sets in data base

[*] Arbitrary units scaled by user-input parameter. Minimum value 0 for no defense. Highest allowed defense level is +7.

[**] Program essential attributes, used and set internally by the QUICK system.

ATTRIBUTE NAME	DESCRIPTION
TYPET*	Target TYPE
TYPE1 TYPE2	Attributes assigned fighter interceptor units (ICLASS = 5 in the data base): attribute TYPE is assigned the TYPE1 or TYPE2 value based on DBMOD input parameter POSTUR (POSTUR = 1 TYPE1 is used; otherwise TYPE2 value used)
VAL	Relative value of an item within its CLASS as established in the data base by the user
VAL1 VAL2	Attributes assigned fighter interceptor units (ICLASS = 5 in the data base): attribute VAL is assigned the VAL1 or VAL2 value based on DBMOD input parameter POSTUR (POSTUR = 1 VAL1 is used; otherwise VAL2 is assigned)
VULN	Vulnerability number
WHDTYP	Warhead type index assigned in the data base
WHTYPN*	Warhead type index (used with EVENTN)
YIELD	Yield (MT)
ZONE	An area bomber defense zone enclosed by a set of linked boundary points

*Program essential attributes, used and set internally by the QUICK system.